REMARKS

Interview Summary

We thank the Examiner for the telephone interview with the associate agents, John Harris (39,465) and Ikuko Wada (43,432) on July 13, 2006.

During the telephone interview, the differences between the present invention and the cited references were discussed, primary Mullins (US Patent 5,857,197). The Examiner kindly acknowledged the differences between the invention and the reference, and suggested that Applicant comment on the definition of the metadata model for clarification.

The subject matter rejection under 35 USC 101 was also discussed. The Examiner kindly suggested amending the claims to include an end result of the transformation.

Claim Amendments

Claims 9-34 and 36-42 are pending. Claims 9 and 36 are independent claims.

Applicant has amended claims 9 and 36 by clarifying that the metadata model constructed by the transformations or steps recited in claims 9 and 36 is stored, as suggested by the Examiner.

Applicant has also corrected a typographical error in the preamble of claim 9.

The Metadata Model

The metadata model stores metadata. Metadata contained in the metadata model is also referred to as "model objects" (page 10, line 34). The metadata model as defined in claims 9 and 36 has three layers of different abstraction. The data access layer 102 is the lowest abstraction layer that contains "a part of the model objects that directly describe actual physical data in the data sources 100 and their relationships" (page 11, lines 16-18). These model objects correspond to the meta data described in Mullins.

The metadata model of the present invention further stores model objects in the business layer 104 and the package layer (106), which are higher abstraction layers. The model objects in the business layer 104 "represent the concepts and structure of the business to be used in business intelligence environments. They represent a single business model, although they can be related to physical data in a number of different data sources" (page 12, lines 14-17). The business model objects "are closely related to the data access model objects in the data access layer 102" (page 12, lines 26-27) as they are constructed based on the data access model objects in the data access layer as recited in claims 9 and 36 and described in the specification, e.g., page 18, lines 10-12.

Similarly, package model objects are constructed based on the business model objects in the business layer as recited in claims 9 and 36 and described in the specification, e.g., page 18, lines 18-20.

Thus, an object in a source database is represented as multiple model objects in multiple layers in the metadata model. The metadata model recited in claims 9 and 36 is "a rich business-oriented metadata model 15 that allows the query engine 30 to generate the best queries of which it is capable, and allows users to build queries ... with the aid of the query engine 30 to obtain desired reports from underlying data sources" (page 8, lines 17-21).

Therefore, the metadata model recited in claims 9 and 36 is very different from the meta data that simply describes data or schema of a data source described in the cited references.

The Invention

The present invention is directed to the transformation of model objects <u>between layers within the metadata model</u>. The metadata model transformer of claim 9 comprises (a) data access model transformations, (b) data access to business model transformations, (c) business model transformations, and (d) business to package model transformations. Especially, <u>the data access to business model transformations</u> constructs business model objects in the business layer based on the data access model objects in the data access layer

by adding business rules for representing business concepts. The business to package model transformations constructs package model objects in the package layer based on the business model objects in the business layer so that the package model objects provide a representation of the business concepts.

Thus, the transformer transforms the model objects from a lower abstraction layer to a higher abstraction layer.

None of the cited references teach or suggest transformation between layers within a metadata model from a lower abstraction layer to a higher abstraction layer.

Applicant trusts that amended claims 9 and 36 and their dependent claims 10-34 and 37-42 are now in condition for allowance.

The following detailed discussions are submitted to fully respond to the Office Action.

Rejection under 35 USC 101

The Examiner has rejected claims 9-34 and 36-42, alleging that these claims are not statutory because (a) they merely recite a number of computing steps without producing any tangible result and/or being limited to a practical application within the technological arts, (b) all the recited steps of the methods or software steps can be done by a person as a metal step or using pencil and paper, and (c) the use of a computer has not been indicated.

Applicant appreciates the Examiner's suggestion during the telephone interview. Applicant trusts that the above amendments to independent claims 9 and 36 have brought all pending claims to comply with the requirements under 35 USC 101.

Rejection under 35 USC 103 against claims 9-21, 24-33 and 36-42

The Examiner rejected claims 9-21, 24-33 and 36-42 being unpatentable over Mullins (USP 5,857,197) in view of Papazoglou et al (A semantic meta-Modeling Approach to schema transformation, 1995) and Fink (USP 6,490,590)

Mullins (US Patent No. 5,857,197)

Mullins does not disclose any metadata model having multiple layers containing model objects of different degrees of abstraction. Thus, Mullins cannot disclose any transformations that can transform objects between layers in a metadata model. Accordingly, Mullins clearly does not teach or suggest data access to business model transformations or business to package model transformations as recited in claim 9.

Mullins discloses a 3-tier architecture of hardware components for accessing data stored in a data store over a distributed network (column 3, line 63). In a 3-tier architecture, the user browser (e.g., object application 101 in Fig. 1) sends a request for data to an application having a business logic (e.g., adapter abstraction layer 600 in Fig. 1), which modifies the request and sends the modified request to a database (data stores 302, 312 in Fig. 1) to retrieve the requested data. The retrieved data is sent back to the application, which in turn modifies and sends the data to the user browser (column 4, lines 9-12; 23-32).

The Examiner alleged that Mulilins discloses data access to business model transformations and business to package model transformations in col. 4, lines 34-48, which reads:

In one embodiment, the subject invention includes the use of meta data 201 (i.e. data used to describe other data) to define how to access and convert non-object data store content 304 to objects and back. This is accomplished by paring down the non-object data store schema 300 into its various components including tables, fields, and conditions 305 in one embodiment. The paring makes the creation, management, and access to the meta data 201 to be a convenient and elegant task. By using the adapter abstraction layer 600 which understands and uses the meta data 201, the subject invention provides an abstract view of an underlying data store(s) 302 (312, 322) as an object store. The effected abstraction produces an architecture whereby the underlying data store(s) 302 (312) as well as multi-tier adapters (e.g. 400, 500, 4XX, and 5XX) may be interchanged without object application 101 code modification.

This section simply describes what Mullins' metadata 201 defines. It does not describe the structure of the metadata 201 or metadata model. Also, this conversion from non-object data store content 304 (e.g., relational data) to objects is performed on the data, not on the metadata. This is not a transformation of meta data 201.

Mullins states that "The adapter abstraction layer 600 of the subject invention performs any translation work necessary for converting objects to both object data stores 312 and non-object data stores 302." (column 4, lines 23-26). Adapter abstraction layer 600 uses meta data 201. The adapter abstraction layer 600 is not an abstraction layer in a metadata mode.

Mullins describes the metadata 201 only as "data used to describe other data" (column 4, line 34). Mullins does not disclose any detail of meta data 201 or any meta data model. Nowhere in Mullins is it disclosed or suggested that meta data 201 has multiple layers. Mullins does not disclose or suggest any data access to business model transformations or business to package model transformations.

The Examiner stated that Mullins does not explicitly teach one or more data access model transformations, and one or more business model transformations. We agree with the Examiner.

Accordingly, Mullins does not disclose or suggest any metadata model transformer, or transformation of objects in a metadata model.

Papazoglou et al (A semantic meta- Modeling Approach to schema transformation, 1995)

The Examiner alleged that Papazoglou discloses one or more data access model transformations for refining description of the physical data in the data source expressed by data access model objects in a metadata model having a data access layer, a business layer and a package layer, as defined in claim 9.

Papazoglou does not disclose any metadata model transformer for transforming a metadata model having multi layers of different abstractions.

Papazoglou describes a mechanism to construct an abstract data model (i.e., an intermediate schema meta-graph (ISMG)), based on a concrete implementation of that data model as represented by a relational database or object-oriented database. This relates to the ability to import metadata from a database/data source and use that metadata to construct a basic data model (ISMG). The importation of metadata to construct a model is different from transformation of object within a model.

In order to construct an abstract data model (ISMG), Papazoglou uses an intermediate meta-model (IMM) which "views a data model as a collection of high-level abstractions (meta-classes) which describe the model's constructs and components, while it views a schema as an instantiation of these abstract constructs" (page 115, left column, first paragraph of section 2.4). The ISMG is an instance of IMM. Thus, the IMM is a meta model which defines the structure of the data model ISMG. The ISMG is a data model that describes the relational or object-oriented database.

The metadata model (15) recited in claim 9 of the present application represents data sources, i.e., describes the data sources. Thus, the metadata model (15) would be considered as a model at the same abstraction level as the ISMG of Papazoglou in terms that both models describe the underlying database/data source. However, these two models are different. The metadata model (15) recited in claim 9 has a data access layer (102), business layer (104) and package layer (106) within the model (15), as exemplified in Figure 2B. The data access model transformation (112) recited in claim 9 refines description of the physical data in the data source expressed by data access model objects in the metadata model (15) having the multiple layers (102, 104, 106).

In contrast, ISMG of Papazoglou has a structure shown in Figure 3 in a single layer, and it does not have multiple layers. Thus, Papazoglou does not disclose any transformation transforming objects from a lower layer to a higher layer within the ISMG.

Papazoglou also describes a mechanism to translate queries against one form of data source (relational or object oriented) into a query against another form of data source, using the data model ISMG (page 113, right column, lines 22-45; page 118, left column, lines 18-30). This translation is an inter-model mapping, and thus it is different from the lower-to-higher transformation within a metadata model as recited in claim 1.

The Examiner cited page 113, left column of Papazoglou. The left column of page 113 describes the translation process of the prior art which "is typically comprised of two steps, first translating from a source (local) data model to a super model and then form the supermodel to a target (remote) model". This is translation between models.

Papazoglou discloses on page 113, right column, lines 8-27, that "The objective of this meta-model is to provide a commonly accepted and understood set of abstractions with which to uniformly represent the semantics of data stored at multiple databases inspite of differences in data representations and data usage" (lines 14-19). The IMM is a meta-model at a higher abstraction level to represent semantics of multiple databases. The IMM and the databases are not two layers within a single metadata model. This section does not describe any transformation in a multi-layered metadata model.

Also, on page 113, right column, lines 22-45, Papazoglou discloses two-step inter-model mapping, i.e., "First by expressing the structures and semantics underlying the heterogeneous models as a series of intermediate meta-classes and then by using instances of these classes in conjunction with a set of translation rules to map from source to target schemas and languages" (lines 36-40). This is about the mapping of data source operations (i.e., query) to another data source using the data model ISMG to perform the translation. This is not transformation in a multi-layered metadata model.

On page 115, left column, Papazoglou describes Figure 2 which shows a hierarchy of meta-classes of the intermediate meta-model IMM. Papazoglou states that "A schema within the IMM is thus represented as instantiations of the meta-classes in the DAG, with DAG nodes representing specializations of the

three broad model-independent categories of DATA-CONSTRUCTS LINKS and CONSTRAINTS". Accordingly, the IMM defines the data model ISMG, and thus, the IMM does not correspond to the metadata model (15) recited in claim 1 which represents data sources. This hierarchy of meta-classes shown in Figure 2 of Papazoglou does not represent different layers of a metadata model. In the first paragraph of section 2.4, Papazoglou describes an instance of the IMM, i.e., the ISMG. An instance (ISMG) of a higher abstraction model (IMM) is not a different layer of the higher abstraction model. Neither sections discloses model objects of different layers in a metadata model. The descriptions on page 115 also do not relate to any metadata model having two layers.

Accordingly, Papazoglou does not disclose or suggest any transformations in a multi-layered metadata model.

Fink (US Patent No. 6,490,590)

The Examiner has referred to the section in Fink that reads "SME refines the business rule metadata to reflect the client's business" (column 8, lines 20-22).

Fink discloses generation of a logical data model and physical data model. These are two separate models.

The step referred to by the Examiner is illustrated as box 326 labeled "modify metadata" in Figure 3B as a step that is carried out after the creation of the physical data model (314) and other steps. This modification of metadata is external modification carried out by a Subject Matter Expert (SME), i.e., a human. Fink does not disclose or suggest use of any transformation to refine business rules. Also, Fink does not disclose how the external modification is carried out.

As shown in Figure 3A, only after the logical data model (LDM) is created (310), a physical data model (PDM) is created (314) "using the GDM tool 300 and the LDM resulting from step 310" (column 6, lines 41-43). In Fink's method, the physical model having a lower degree of abstraction is created after the logical model having a higher degree of abstraction is created. This is totally

opposite to the transformations carried out by the metadata model transformer of the present invention, as recited in independent claims 9 and 36.

As discussed above, While Fink discloses modification of metadata, it is external modification carried out by a Subject Matter Expert (SME), i.e., a human, Fink does not disclose or suggest use of any transformation to refine business rules, nor how modification is done.

Fink does not disclose any transformation which can construct model objects of a higher degree of abstraction based on model objects of a lower degree of abstraction.

Therefore, even if one skilled in the art combines Mullins, Papazoglou and Fink, he would use a metadata model with no layers as per Mullins or Papazoglou, and attempts to modify the metadata model as per Fink to create a new model of a lower degree of abstraction. He would still fail to provide a metadata model transformer or method for transforming a metadata model as recited in claims 9 and 36 and their dependent claims.

Consequently, the present invention as recited in claims 9-21, 24-33 and 36-42 have been patentably distinguished over Mullins, Papazoglou and Fink.

Rejection under 35 USC 103 against claims 22 and 23

The Examiner rejected claims 22 and 23 being unpatentable over Mullins in view of Fink and Henninger et al (USP 5,499,371).

Claims 22 and 23 depend on claim 21 which depends on claim 9.

Henninger et al (US Patent NO. 5,499,371)

Henninger et al discloses an apparatus for using an object model of an object-oriented application to automatically map information between an object-oriented application and a structured database, e.g., a relational database. As described on column 8, lines 48-53, for each one-to-one and one-to-many relationship in the object model, a foreign key column or foreign key columns are added to the database table schema in the appropriate table of the database schema, and for each many-to-many relationship in the object model, a separate

join table is added to the database schema. In these cases, Henninger's method constructs a database schema and a transform, using the object model as input.

The object model of Henninger is not transformed. As shown in Figures 1 and 3, Henninger's method software 15 accepts object model 20 and accepts or creates database schema 30 and transform 50, and using these three elements as input, the method automatically generates source code (column 5, lines 63-65; column 7, lines 29-31). As seen in Step D of Figure 3, Henninger's method constructs a database schema 30 and a transform 50 derived from the object model 20 (column 7, lines 53-55). Thus, Henninger simply uses the input object model 20, and does not modify or transform the object model 20 as part of the process. This is contrary to the present invention that transforms the metadata model. Therefore, the present invention as claimed is totally different from Henninger.

Therefore, it is respectfully submitted that claims 22 and 23 are also patentably distinguished over Mullins, Fink and Henninger.

CONCLUSION

Having dealt with all rejections, Applicant trusts that the application is now in condition for allowance. Early favorable reconsideration of the application is respectfully requested. If the Examiner requires any further clarification or has any questions, we respectfully requests the Examiner to contact Applicant's associate agent John Harris at 613-786-8671.

Respectfully submitted,
GARDNER GROFF SANTOS &
GREENWALD, PC

/djs/
Daniel J. Santos
Reg. No. 40,158

Customer No. 23506
GARDNER GROFF SANTOS & GREENWALD, PC
2018 Powers Ferry Road, Suite 800
Atlanta, Georgia 30339
Phone: 770.984.2300
Fax: 770.984.0098